1781-PXB241
User's Manual

Western Reserve Controls, Inc.

Although every effort has been made to insure the accuracy of this document, all information is subject to change without notice. Western Reserve Controls, Inc. assumes no liability for any errors or omissions in this document or for direct, indirect, incidental or consequential damage resulting from the use of this document.

Document PUB 19.0
Rev 1.60
January 2000

Copyright © 1992-2000 WRC

Western Reserve Controls, Inc.
1485 Exeter Road
Akron OH 44306
330-733-6662 (Phone)
330-733-6663 (Fax)
sales@wrcakron.com (Email)
http://www.wrcakron.com (Web)

WRC is a trademark of Western Reserve Controls, Inc.
All other trademarks are property of their respective companies.

TABLE OF CONTENTS

INTRODUCTION **1**
BOARD DIMENSIONS **2**
HARDWARE ADDRESSING **3**
HARDWARE INSTALLATION **3**
SOFTWARE **4**
COMMUNICATING WITH THE BOARD / SOFTWARE ADDRESSING **4**
INPUT/OUTPUT INITIALIZATION **5**
ENVIRONMENTAL SPECIFICATIONS **6**
CABLES AND CONNECTORS **6**
TROUBLESHOOTING **8**
EXAMPLE BASIC PROGRAM **9**
EXAMPLE C PROGRAM **11**
APPENDIX A **12**

Western Reserve Controls, Inc.
Akron, Ohio

INTRODUCTION

The 1781-PXB241 board provides a direct interface between the personal computer (PC) and a digital I/O rack for single point I/O modules. The 1781-PXB241 board is plugged into the PC backplane and a 50-pin ribbon cable is connected between the 1781-PXB241 and the I/O rack / mounting board. The 1781-PXB241 monitors and/or controls up to 24 individual I/O points on a single rack.

The 1781-PXB241 is a single half-slot board that plugs directly into a IBM or IBM-compatible personal computer XT or AT bus (PC bus). The 1781-PXB241 is compatible with the Opto22 AC5 and allows the user to connect any standard 50-pin I/O rack to the 1781-PXB241 interface board. The connection from the 1781-PXB241 interface to the I/O rack is through a 50 pin flat-ribbon cable. The 1781-PXB241 can be used with Western Reserve Controls line of industrial digital I/O modules and mounting boards, as well as those from other manufacturers, including standard size, miniature, or new G4/G5 I/O modules.

- Up to 24 digital I/O points per 1781-PXB241
- Compatible with the Opto22 AC5
- PC XT or AT bus, half slot
- Selection of I/O address
- Dip switch for I/O address
- 5 V power to I/O mounting rack optionally supplied

BOARD DIMENSIONS

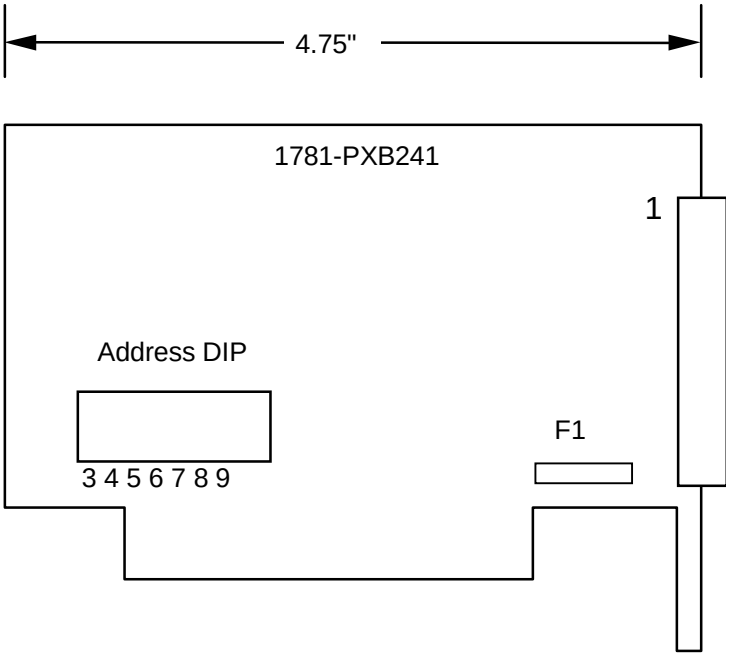

Figure 1

HARDWARE ADDRESSING

The 1781-PXB241 board is mapped into the I/O space of the personal computer. All communications between the PC and the 1781-PXB241 are done using input (INP) and output (OUTP) computer instructions. The addresses of the INP and OUTP instructions are determined by setting the DIP switch on the 1781-PXB241 board. The base address of the 1781-PXB241 is determined by the DIP switch setting. The base address of 1781-PXB241 must be selected prior to installation of the 1781-PXB241 board. Below is a description of the DIP switch and jumper settings:

(ON = CLOSED = 0, OFF = OPEN = 1)

BASE ADDRESS	A3	A4	A5	A6	A7	A8	A9	Not Used
220H	**ON**	**ON**	**OFF**	**ON**	**ON**	**ON**	**OFF**	**OFF**
230H	**ON**	**OFF**	**OFF**	**ON**	**ON**	**ON**	**OFF**	**OFF**
240H	**ON**	**ON**	**ON**	**OFF**	**ON**	**ON**	**OFF**	**OFF**
250H	**ON**	**OFF**	**ON**	**OFF**	**ON**	**ON**	**OFF**	**OFF**
260H	**ON**	**ON**	**OFF**	**OFF**	**ON**	**ON**	**OFF**	**OFF**
280H	**ON**	**ON**	**ON**	**ON**	**OFF**	**ON**	**OFF**	**OFF**
290H	**ON**	**OFF**	**ON**	**ON**	**OFF**	**ON**	**OFF**	**OFF**
290H	**OFF**	**ON**	**OFF**	**ON**	**ON**	**OFF**	**ON**	**OFF**
2A0H	**ON**	**ON**	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**OFF**
2B0H	**ON**	**OFF**	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**OFF**
3E0H (Default)	**ON**	**ON**	**OFF**	**OFF**	**OFF**	**OFF**	**OFF**	**OFF**

HARDWARE INSTALLATION

1. Set the address DIP switch to a desired address. The address DIP switch is shown in Figure 1. Select a base address for the 1781-PXB241 and set the individual DIP switches to ON or OFF corresponding to the selected base address. The base address and switch settings are shown above.

2. The 1781-PXB241 can supply 5 volt logic power to the I/O mounting board (rack) via line 49 in the 50-conductor ribbon cable. Determine if the power to the I/O mounting board is to be supplied from an external power source or from the P1781-PXB241. If power is to be supplied from the computer and PXB241, then make sure the 5 V fuse jumper F1 is installed.

3. If the 5 V power for the I/O mounting board is to be supplied from an external power source, then **F1 must be removed**.

> **Important**: If an external power supply is used and F1 is not removed from the PXB241, serious damage to the computer and the PXB241 may result.

4. With power OFF, place the 1781-PXB241 board into an open slot inside your personal computer (PC). Screw the board down.

5. Attach one end of a 50-pin ribbon cable to the 50 pin header of the 1781-PXB241 board. Attach the opposite end of the 50 pin ribbon cable to the I/O rack. Be sure to align the pin 1's of the ribbon cable. Most ribbon cables are keyed or have a red line or wire as pin 1.

6. Turn on the personal computer.

7. If F1 was removed from the PXB241, connect and turn on the power to the I/O rack.

8. To test the board and I/O rack, execute the sample basic program listed in this manual.

9. If you have problems, see the troubleshooting section of this manual.

The 1781-PXB241 comes with a support diskette. As with any software package, you should make backup copies for everyday use and place your original master diskette in a safe location.

To install software on your hard drive do the following:

1) Place the software diskette in drive.
2) Change DOS prompt to appropriate drive (usually A:).
3) Type INSTALL and follow the screen prompts.

Files contained on the diskette are stored in separate directories as follows:

SETUP: This directory contains FINDBASE.EXE and WRC.EXE. These DOS programs provide assistance finding an open memory address and setting jumpers on the 1781-PXB241 card.

TEST: This directory contains a basic sample and executable for use with the 1781-PXB241.

PXB241: This directory contains basic and C samples as well as a DOS setup program (PXB241.exe)

VB_WRC: This directory contains a sample VisualBASIC 3 for Windows program as well as a VisualBASIC linkable driver for 16-bit operation. (see APPENDIX A).

COMMUNICATING WITH THE BOARD / SOFTWARE ADDRESSING

The 1781-PXB241 uses a consecutive block of 8 I/O addresses. Each one of these I/O addresses corresponds to a control or data register. Once the 1781-PXB241 base address is determined, this sets the first of 8 I/O addresses to be used to communicate with each of the 3 sets of 8 bit I/O banks.

I/O ADDRESS	REGISTER	I/O BITS

BASE + 0 BASE + 1	DATA1 CONTROL 1	0 - 7
BASE + 2 BASE + 3	DATA2 CONTROL 2	8 - 15
BASE + 4 BASE + 5	DATA3 CONTROL 3	16 - 23

INPUT/OUTPUT INITIALIZATION

The 1781-PXB241 uses 6 of the 8 registers for control and data of the individual I/O points. The 1781-PXB241 must be initialized first prior to its use. Each point must be configured as an input or output channel. This is accomplished by writing a value to the configuration or control register. The following procedure describes the 1781-PXB241 initialization:

- Output a zero (0) to each control register
- Output the configuration byte to each data register. A "1" bit in the configuration byte represents an output, and a "0" bit represents an input.
- Output a value of 34 hex to the control register.
- Now read or write the data register to read or write values to each of the 8 bit I/O banks.

It is important to note that negative logic is used for outputs. Writing a "1" bit to an output module will turn the module off, and writing a "0" bit will turn the module on.

- "1" is output OFF
- "0" is output ON

The read inputs also use negative logic. Reading a "1" means the input is off, and reading a "0" means the input is on.

- "1" is input OFF
- "0" is input ON

ENVIRONMENTAL SPECIFICATIONS

Temperature	0 to 60°C
Relative Humidity	0 to 95% non-condensing
International Agency Approvals	CE EMC (Electromagnetic Compatibility)

CABLES AND CONNECTORS

1781-PXB241 is connected to an I/O rack via a 50-conductor, flat ribbon cable. The standard cable available from WRC is:

1781-C2EH	a cable of 2 feet in length, with a header connector at one end and a card-edge connector at the other

Other cable lengths are available by substituting the "2" with the desired length, yielding a catalog number of the form "1781-CxEH".

To make your own cables, use the following cables and connectors, or their equivalents:

Cables:

Alpha Part #	3M Part #	Type
3580/50 or 3583/50	3365/50	Regular
3584/50 or 3476/50	3353/50 or 3469/50	Ground Plane
3589/50	3603/50	Jacketed
3590/50	3517/50	Jacketed and Gnd Plane

Connectors:

Manufacturer	Part #	Type
3M	3425-7000	Header
Circuit Assembly	CA-50IDSB	Header
3M	3415-0001	Edge card

TROUBLESHOOTING

The 1781-PXB241 test and configuration software is available in the SETUP and can help you configure the jumpers on the 1781-PXB241. The 1781-PXB241 software also allows the user to test the individual I/O points. Use the TEST program or the VisualBASIC for Windows sample to test I/O.

The following are some helpful hints:

- Check the 5 volt supply at the I/O rack.
- Check the base address of the 1781-PXB241 board, ON = closed, OFF = open.
- Check that pin 1 of the ribbon goes to pin 1 on the rack.
- Write a program to turn on/off a combination of every other output point. The example below was used on an IBM AT, with a 1781-PXB241 and a 24-point I/O rack with all outputs.

EXAMPLE BASIC PROGRAM

```
100'.....
110'  This example is for an IBM PC/XT/AT and a 1781-PXB241 24-Point I/O PC
120'  Card addressed at 3E0 hex.  It sets up the first 16 I/O as outputs and turns them
130'  all OFF.  It sets up the last 8 I/O as inputs and reads them and prints the status.
140'.....
150'  BASE% = &H3E0
160'.....
170'  Clear the Control Registers
180'.....
190  OUT BASE% + 1, 0
200  OUT BASE% + 3, 0
210  OUT BASE% + 5, 0
220'.....
230'  Set up first 2 ports as all Outputs / 1=output, 0=input
240'.....
250  OUT BASE% + 0, &HFF
260  OUT BASE% + 2, &HFF
270  OUT BASE% + 4, &H00
280'.....
290'  Set up last port as all Inputs
300'.....
310'.....
320'  Set up the Control Registers / Always &H34
330'.....
340  OUT BASE% + 1, &H34
350  OUT BASE% + 3, &H34
360  OUT BASE% + 5, &H34
370'.....
380'  Turn all Outputs Off / 1 = OFF, 0=ON
390'.....
400  OUT BASE% + 0, &HFF
410  OUT BASE% + 2, &HFF
420'.....
430'  Read the Inputs
440'.....
450  X% = INP(BASE% + 4)
460  PRINT "PORT C INPUT STATUS = ", X%
470  PRINT "BIT  = 1 => OFF; BIT = 0 => ON
480'
```

```
490  END
```

EXAMPLE C PROGRAM

```
/*  This example is for an IBM PC/XT/AT and a 1781-PXB241 addressed at 3E0H.
It sets up the first 16 I/O as outputs and turns them all OFF.  It sets up the last 8 I/O
as inputs and reads them and prints their status.
*/
main () {
        int base;

        /* set the base address */
        base = 0x3e0;

        /* clear the control registers */
        outp(base+1,0);
        outp(base+3,0);
        outp(base+5,0);

        /*set up first 2 ports as all Outputs / 1=output, 0=input, 3rd port as input */
        outp(base+0,0xff);
        outp(base+2,0xff);
        outp(base+4,0x00);

        /*set up the Control Registers / Always 0x34 */
        outp(base +1,0x34);
        outp(base +3,0x34);
        outp(base +5,0x34);

        /* turn all Outputs Off / 1 =off, 0=on */
        outp(base+0,0xff);
        outp(base+2,0xff);
        outp(base+4,0xff);}
        /* read the inputs  and prints their status*/
        status = inportp(base+4,0xff);
        printf ("\n\nThe input status of port C is: %x", status);
}
```

APPENDIX A

VISUALBASIC UTILITY DRIVER

Western Reserve Controls provides extensions to the VisualBASIC language on the diskette provided with your card. The extensions are in a directory named VB_WRC. These extensions are in the form of a .DLL, .GBL, and a VisualBASIC 3.0 sample. Together these files allow you to access the port and main memory space in a fashion similar to BASIC, QuickBASIC, Pascal, C/C++, Assembly, and most other standard languages.

To use these files in a VisualBASIC program, you must create a .MAK file (File | New Project) similar to the sample provided (or else modify your existing project file) and include the .GBL file (File | Add File). Once this has been done, VisualBASIC will be enhanced with the addition of the following functions. Note that VB_WRC.DLL must be in the path stated in the .GBL file for any of the following functions to work..

Release 1.21 of the PXB-241 contains the "pxb95.dll" file that allows it to be used in a Windows95 and VisualBASIC 5.0 environment. This version was written to support 32-bit mode of operation in Windows95.

You must replace the references to "vb_wrc.dll" with "pxb95.dll" in the examples shown for VB3 in order to use it with VB5.0.

Inportb

Function: Reads a byte from a hardware port. Due to limitations of VisualBASIC, the number is returned as an integer.

Declaration: function InPortb (byval *address* as integer) as integer

InPort

Function: Reads an integer from a hardware port. This function returns the 16-bit value obtained from reading the low byte from address and the high byte from *address+1.*

Declaration: function InPort (byval *address* as integer) as integer.

OutPortb

Function: Writes the lower eight bits of *value* to the hardware at port *address.* This returns the value output.

Declaration: function OutPortb(byval *address* as integer, byval *value* as integer) as integer.

OutPort

Function: Writes all 16 bits of *value* to the hardware port at *address*. This function returns the value output.

Declaration: function OutPort(byval *address* as integer, byval *value* as integer) as integer.

Peek

Function: Reads a byte form main memory (DRAM).

Declaration: function Peek(byval segment as integer, byval offset as integer) as integer.

Poke

Function: Writes the lower eight bits of *value* to *segment:offset*.

Declaration: function Poke(byval segment as integer, byval offset as integer, byval value as integer) as integer.

Note that in all the above functions, an inherent limitation of BASIC in general and VisualBASIC in particular makes the values sent less intuitive. All integers in BASIC are signed numbers, wherein data are stored in two's complement form. All bit patterns must be converted to-and-from this two's complement form if meaningful display is required. Otherwise, values returned from the InPortb function will be -128 to 127, rather than 0 to 255. An alternative is to perform all assignments in hexadecimal, rather than decimal form.

Before the program will execute, the .GBL file must be modified to include the path to the VB_WRC.DLL as appropriate for your system. Merely replace the statement "VB_WRC.DLL" with "*drive:path* \VB_WRC.DLL".

As an alternative to changing the source code, you can copy the VB_WRC file into your Windows directory. This will allow multiple programs to find the same .DLL without having to know where it is located. Just leave off all references to a path in the .GBL file as shown in the sample.